

CS477 Formal Software Dev Methods

Elsa L Gunter
2112 SC, UIUC
egunter@illinois.edu
<http://courses.engr.illinois.edu/cs477>

Slides based in part on previous lectures
by Mahesh Vishwanathan, and by Gul Agha

February 26, 2020

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

1 / 1

Substitution in Formulae: Two Approaches

- When quantifier would capture free variable of redex, can't substitute in formula as is
- Solution 1: Make substitution partial function – undefined in this case
- Solution 2: Define equivalence relation based on renaming bound variables; define substitution on equivalence classes
- Will take Solution 1 here
- Still need definition of equivalence up to renaming bound variables

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

2 / 1

Substitution in Formulae

- Defined by structural induction; uses substitution in terms
- Read equations below as saying left is not defined if any expression on right not defined
- $\text{true}[t/x] = \text{true}$ $\text{false}[t/x] = \text{false}$
- $r(t_1, \dots, t_n)[t/x] = r((t_1[t/x], \dots, t_n[t/x]))$
- $(\psi)[t/x] = (\psi[t/x])$ $(\neg\psi)[t/x] = \neg(\psi[t/x])$
- $(\psi_1 \otimes \psi_2)[t/x] = (\psi_1[t/x]) \otimes (\psi_2[t/x])$ for $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $(\mathcal{Q}x. \psi)[t/x] = \mathcal{Q}x. \psi$ for $\mathcal{Q} \in \{\forall, \exists\}$
- $(\mathcal{Q}y. \psi)[t/x] = \mathcal{Q}y. (\psi[t/x])$ if $x \neq y$ and $y \notin \text{fv}(t)$ for $\mathcal{Q} \in \{\forall, \exists\}$
- $(\mathcal{Q}y. \psi)[t/x]$ not defined if $x \neq y$ and $y \in \text{fv}(t)$ for $\mathcal{Q} \in \{\forall, \exists\}$

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

3 / 1

Substitution in Formulae

Examples

$(x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[x + y/z]$ not defined

$(x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee (z \geq w)))[x + y/z] =$
 $(x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee ((x + y) \geq y)))$

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, formula ψ over \mathcal{G} , and a assignment. If $\psi[t/x]$ defined, then $a \models^{\mathcal{S}} \psi[t/x]$ if and only if $a[x \mapsto \mathcal{T}_a(t)] \models^{\mathcal{S}} \psi$

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

4 / 1

Renaming by Swapping: Terms

Define the **swapping** of two variables in a term $t[x \leftrightarrow y]$ by structural induction on terms:

- $x[x \leftrightarrow y] = y$ and $y[x \leftrightarrow y] = x$
- $z[x \leftrightarrow y] = z$ for z a variable, $z \neq x, z \neq y$
- $f(t_1, \dots, t_n)[x \leftrightarrow y] = f(t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y])$

Examples:

$\text{add}(1, \text{abs}(\text{add}(x, y)))[x \leftrightarrow y] = \text{add}(1, \text{abs}(\text{add}(y, x)))$
 $\text{add}(1, \text{abs}(\text{add}(x, y)))[x \leftrightarrow z] = \text{add}(1, \text{abs}(\text{add}(z, y)))$

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

5 / 1

Renaming by Swapping: Terms

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, variables x and y , term t over \mathcal{G} , and a assignment. Let $b = a[x \mapsto a(y)][y \mapsto a(x)]$. Then $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$

Elsa L. Gunter

CS477 Formal Software Dev Methods

February 26, 2020

6 / 1

Renaming by Swapping: Terms

Proof.

By structural induction on terms, suffices to show theorem for the case where t variable, and case $t = f(t_1, \dots, t_n)$, assuming result for t_1, \dots, t_n

- Case: t variable

- Subcase: $t = x$. Then $\mathcal{T}_a(x[x \leftrightarrow y]) = \mathcal{T}_a(y) = a(y)$ and $\mathcal{T}_b(x) = b(x) = a[x \mapsto a(y)][y \mapsto a(x)](x) = a[x \mapsto \mathcal{T}_a(y)](x) = a(y)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$
- Subcase: $t = y$. Then $\mathcal{T}_a(y[x \leftrightarrow y]) = \mathcal{T}_a(x) = a(x)$ and $\mathcal{T}_b(y) = b(y) = a[x \mapsto a(y)][y \mapsto a(x)](x) = a(x)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$
- Subcase: $t = z$ variable, $z \neq x$ and $z \neq y$. Then $\mathcal{T}_a(z[x \leftrightarrow y]) = \mathcal{T}_a(z) = a(z)$ and $\mathcal{T}_b(z) = b(z) = a[x \mapsto a(y)][y \mapsto a(x)](z) = a[x \mapsto \mathcal{T}_a(y)](z) = a(z)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$

Renaming by Swapping: Terms

Proof.

- Case: $t = f(t_1, \dots, t_n)$. Assume $\mathcal{T}_a(t_i[x \leftrightarrow y]) = \mathcal{T}_b(t_i)$ for $i = 1, \dots, n$. Then

$$\begin{aligned} \mathcal{T}_a(t[x \leftrightarrow y]) &= \mathcal{T}_a(f(t_1, \dots, t_n)[x \leftrightarrow y]) \\ &= \mathcal{T}_a(f(t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y])) \\ &= \phi(f)(\mathcal{T}_a(t_1[x \leftrightarrow y]), \dots, \mathcal{T}_a(t_n[x \leftrightarrow y])) \\ &= \phi(f)(\mathcal{T}_b(t_1), \dots, \mathcal{T}_b(t_n)) \\ &\quad \text{since } \mathcal{T}_a(t_i[x \leftrightarrow y]) = \mathcal{T}_b(t_i) \text{ for } i = 1, \dots, n \\ &= \mathcal{T}_b(f(t_1, \dots, t_n)) \\ &= \mathcal{T}_b(t) \quad \square \end{aligned}$$

Renaming by Swapping: Formulae

Define the **swapping** of two variables in a formula $\psi[x \leftrightarrow y]$ by structural induction, using swapping on terms:

- $\text{true}[x \leftrightarrow y] = \text{true}$ $\text{false}[x \leftrightarrow y] = \text{false}$
- $r(t_1, \dots, t_n)[x \leftrightarrow y] = r(t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y])$
- $(\psi)[x \leftrightarrow y] = (\psi[x \leftrightarrow y])$ $(\neg\psi)[x \leftrightarrow y] = \neg(\psi[x \leftrightarrow y])$
- $(\psi_1 \otimes \psi_2)[x \leftrightarrow y] = (\psi_1[x \leftrightarrow y]) \otimes (\psi_2[x \leftrightarrow y])$ for $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $(Qx. \psi)[x \leftrightarrow y] = Qy. (\psi[x \leftrightarrow y])$ for $Q \in \{\forall, \exists\}$
- $(Qy. \psi)[x \leftrightarrow y] = Qy. (\psi[x \leftrightarrow y])$ for $Q \in \{\forall, \exists\}$
- $(Qz. \psi)[x \leftrightarrow y] = Qz. (\psi[x \leftrightarrow y])$ for z a variable with $z \neq x$, $z \neq y$, and $Q \in \{\forall, \exists\}$

Renaming by Swapping: Formulae

Examples

$$\begin{aligned} (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[x \leftrightarrow y] \\ &= (y > 3 \wedge (\exists x. (\forall z. z \geq (x - y)) \vee (z \geq x))) \\ (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[y \leftrightarrow z] \\ &= (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[y \leftrightarrow w] \end{aligned}$$

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, variables x and y , formula ψ over \mathcal{G} , and a assignment. If $x \notin \text{fv}(t)$ and $y \notin \text{fv}(t)$ then $\psi[x \leftrightarrow y] \equiv \psi$

α -equivalence

- $\psi \stackrel{\alpha}{\equiv} \psi$
- If $\psi_1 \stackrel{\alpha}{\equiv} \psi_2$ then $\psi_2 \stackrel{\alpha}{\equiv} \psi_1$.
- If $\psi_1 \stackrel{\alpha}{\equiv} \psi_2$ and $\psi_2 \stackrel{\alpha}{\equiv} \psi_3$ then $\psi_1 \stackrel{\alpha}{\equiv} \psi_3$
- If $x \notin \text{fv}(\psi)$ and $y \notin \text{fv}(\psi)$ then $\psi \stackrel{\alpha}{\equiv} \psi[x \leftrightarrow y]$.
- If $\psi_i \stackrel{\alpha}{\equiv} \psi'_i$ for $i = 1, 2$ then
 - $(\psi_1) \stackrel{\alpha}{\equiv} (\psi'_1)$ $\neg\psi_1 \stackrel{\alpha}{\equiv} \neg\psi'_1$
 - $\psi_1 \otimes \psi_2 \stackrel{\alpha}{\equiv} \psi'_1 \otimes \psi'_2$ for $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
 - $Qz. \psi_1 \stackrel{\alpha}{\equiv} Qz. \psi'_1$ for $Q \in \{\forall, \exists\}$

α -equivalence: Example

$$\begin{aligned} (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y))) \\ &\stackrel{\alpha}{\equiv} (x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee (z \geq w))) \end{aligned}$$

$$\begin{aligned} (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y))) \\ &\stackrel{\alpha}{\equiv} (x > 3 \wedge (\exists w. (\forall y. y \geq (w - x)) \vee (z \geq w))) \end{aligned}$$

Proof Rules

Will give Sequent version of Natural Deduction rules
All rules from Propositional Logic included

$$\frac{\Gamma \vdash \psi'[t/x]}{\Gamma \vdash \exists x.\psi} \text{ Ex I} \quad \text{provided } \psi \stackrel{\alpha}{\equiv} \psi'$$

$$\frac{\Gamma \vdash \exists x.\psi \quad \Gamma \cup \{(\psi[y/x])\} \vdash \varphi}{\Gamma \vdash \varphi} \text{ Ex E}$$

provided $y \notin \text{fv}(\varphi) \cup (\text{fv}(\psi) \setminus \{x\}) \cup \bigcup_{\psi' \in \Gamma} \text{fv}(\psi')$

$$\frac{\Gamma \vdash \psi[y/x]}{\Gamma \vdash \forall x.\psi} \text{ All I}$$

provided $y \notin (\text{fv}(\psi) \setminus \{x\}) \cup \bigcup_{\psi' \in \Gamma} \text{fv}(\psi')$

$$\frac{\Gamma \vdash \forall x.\psi \quad \Gamma \cup \{\psi'[t/x]\} \vdash \varphi}{\Gamma \vdash \varphi} \text{ All E}$$

provided $\psi \stackrel{\alpha}{\equiv} \psi'$

Example

Show

$$\frac{}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)}$$

Example

Show

$$\frac{\frac{}{\{\exists x.\forall y. x \leq y\} \vdash \forall x.\exists y. y \leq x}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}$$

Example

Show

$$\frac{\frac{\frac{\{\exists x.\forall y. x \leq y\} \vdash \exists y. y \leq x}{\{\exists x.\forall y. x \leq y\} \vdash \forall x.\exists y. y \leq x} \text{ All I}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}$$

Example

Show

$$\frac{\frac{\frac{\{\exists x.\forall y. x \leq y\} \vdash \exists x.\forall y. x \leq y \quad \left\{ \begin{array}{l} \exists x.\forall y. x \leq y; \\ \forall y.z \leq y \end{array} \right\} \vdash \exists y. y \leq x}{\{\exists x.\forall y. x \leq y\} \vdash \exists y. y \leq x} \text{ Ex E}}{\frac{\frac{\{\exists x.\forall y. x \leq y\} \vdash \forall x.\exists y. y \leq x}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}} \text{ All I}}$$

Example

Show

$$\frac{\frac{\frac{\frac{\{\exists x.\forall y. x \leq y\} \vdash \exists x.\forall y. x \leq y}{\{\exists x.\forall y. x \leq y\} \vdash \exists y. y \leq x} \text{ Hyp}}{\frac{\frac{\{\exists x.\forall y. x \leq y\} \vdash \forall x.\exists y. y \leq x}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ All I}} \text{ Ex E}}{\{\} \vdash (\exists x.\forall y. x \leq y) \Rightarrow (\forall x.\exists y. y \leq x)} \text{ Imp I}}$$

Floyd-Hoare Logic

- Also called **Axiomatic Semantics**
- Based on formal logic (first order predicate calculus)
- Logical system built from **axioms** and **inference rules**
- Mainly suited to simple imperative programming languages
- Ideas applicable quite broadly

Floyd-Hoare Logic

- Used to formally prove a property (**post-condition**) of the **state** (the values of the program variables) after the execution of program, assuming another property (**pre-condition**) of the state holds before execution

Floyd-Hoare Logic

- Goal: Derive statements of form

$$\{P\} C \{Q\}$$

- P, Q logical statements about state, P precondition, Q postcondition, C program
- Example:

$$\{x = 1\} x := x + 1 \{x = 2\}$$

Floyd-Hoare Logic

- **Approach:** For each type of language statement, give an axiom or inference rule stating how to derive assertions of form

$$\{P\} C \{Q\}$$

where C is a statement of that type

- Compose axioms and inference rules to build proofs for complex programs

Partial vs Total Correctness

- An expression $\{P\} C \{Q\}$ is a **partial correctness** statement
- For **total correctness** must also prove that C terminates (i.e. doesn't run forever)
 - Written: $[P] C [Q]$
- Will only consider partial correctness here

Simple Imperative Language

- We will give rules for simple imperative language

```
 $\langle \text{command} \rangle ::= \langle \text{variable} \rangle := \langle \text{term} \rangle$   
|  $\langle \text{command} \rangle; \dots; \langle \text{command} \rangle$   
| if  $\langle \text{statement} \rangle$  then  $\langle \text{command} \rangle$  else  $\langle \text{command} \rangle$   
| while  $\langle \text{statement} \rangle$  do  $\langle \text{command} \rangle$ 
```

- Could add more features, like for-loops

Substitution

- Notation: $P[e/v]$ (sometimes $P[v \rightarrow e]$)
- Meaning: Replace every v in P by e
- Example:

$$(x + 2)[y - 1/x] = ((y - 1) + 2)$$

The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{ \quad ? \quad \} x := y \{x = 2\}}$$

The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{\square = 2\} x := y \{\square = 2\}}$$

The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{\square = 2\} x := y \{\square = 2\}}$$

The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Examples:

$$\frac{}{\{y = 2\} x := y \{x = 2\}}$$

$$\frac{}{\{y = 2\} x := 2 \{y = x\}}$$

$$\frac{}{\{x + 1 = n + 1\} x := x + 1 \{x = n + 1\}}$$

$$\frac{}{\{2 = 2\} x := 2 \{x = 2\}}$$

The Assingment Rule – Your Turn

- What is the weakest precondition of

$$x := x + y \{x + y = wx\}$$

$$\frac{}{\{ \quad ? \quad \} x := x + y \{x + y = wx\}}$$

The Assignment Rule – Your Turn

- What is the weakest precondition of

$$x := x + y \{x + y = wx\}?$$

$$\frac{\{ (x + y) + y = w(x + y) \}}{x := x + y \{x + y = wx\}}$$

Precondition Strengthening

$$\frac{(P \Rightarrow P') \{P'\} C \{Q\}}{\{P\} C \{Q\}}$$

- Meaning: If we can show that P implies P' (i.e. $P \Rightarrow P'$) and we can show that $\{P'\} C \{Q\}$, then we know that $\{P\} C \{Q\}$
- P is **stronger** than P' means $P \Rightarrow P'$

Precondition Strengthening

- Examples:

$$\frac{x = 3 \Rightarrow x < 7 \quad \{x < 7\} x := x + 3 \{x < 10\}}{\{x = 3\} x := x + 3 \{x < 10\}}$$

$$\frac{True \Rightarrow (2 = 2) \quad \{2 = 2\} x := 2 \{x = 2\}}{\{True\} x := 2 \{x = 2\}}$$

$$\frac{x = n \Rightarrow x + 1 = n + 1 \quad \{x + 1 = n + 1\} x := x + 1 \{x = n + 1\}}{\{x = n\} x := x + 1 \{x = n + 1\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}{\{x = 3\} x := x * x \{x < 25\}}$$

$$\frac{\{x = 3\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}$$

$$\frac{\{x * x < 25\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}{\{x = 3\} x := x * x \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}$$

$$\frac{\{x * x < 25\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}{\{x = 3\} x := x * x \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}} \text{ NO}$$

$$\frac{\{x * x < 25\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}}{\{x = 3\} x := x * x \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}} \text{ NO}$$

$$\frac{\{x * x < 25\} x := x * x \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \{x < 25\}} \text{ YES}$$